ON FAULT IDENTIFICATION OF LINEAR NETWORKS

by

James Richard Dyer

United States
Naval Postgraduate School

THESIS

ON FAULT IDENTIFICATION OF
LINEAR NETWORKS

by

James Richard Dyer

June 1969

On Fault Identification of

Linear Networks

by

James Richard Dyer
Lieutenant (junior grade), United States Navy
B.S.E.E., University of Arkansas, 1968

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 1969

ABSTRACT

A study has been made on the possibility of identifying faulty components of a network by making tests only on external points of that network. The conclusion is made that this is possible, and that a practical method of doing this can be developed from the computer program presented as a result of this investigation.

The method used is to select the voltage transfer functions of a network as the quantities on which the tests will be made. The poles and zeros of these functions are used to select a set of test frequencies. From the measurements made at these frequencies, a set of signatures is available which allow the faulty components to be identified.

TABLE OF CONTENTS

ACKNOWLEDGMENT

The author wishes to thank Professor Shu-Gar Chan for his guidance during the period of research and the preparation of this thesis.

I. INTRODUCTION

One of the urgent needs facing today's Navy is a solution to the problem of electric and electronic equipment failures. It is not likely that such highly reliable components will be produced that the problem can be eliminated. Therefore, another approach is necessary.

In the event of an equipment failure in an emergency situation, many lives may depend on how fast the repairs are made. Often the only method available to the technician is to disassemble the equipment and make tests at various internal points. When the technician does not have a good familiarity with the equipment, this can lead to a down time of up to several hours, which in most cases is intolerable. Therefore, a method of maintenance is needed which would allow even an inexperienced technician to have the repairs completed in the shortest possible time. Also, it would be desirable to be able to apply this same procedure to a routine program of preventive maintenance to eliminate these failures before they can occur.

A logical approach to this problem would be to devise a scheme that would use only external test points of the equipment (i.e., the input and/or output terminals). Such an approach has been used in this investigation. Some previous studies in this area have been made [1], [2].

This investigation has been carried out in the form of a feasibility study rather than to develop a practical method. The approach used was to select the voltage transfer functions of a network as the quantities on which tests are to be made. The considerations leading to this decision are given in Chapter II.

Chapters III and IV cover the method of selecting test frequencies and a method of assigning signatures to the results obtained from these tests. Chapter V and the COMPUTER OUTPUT section cover the results of this investigation.

II. PRELIMINARY CONSIDERATIONS

A. GENERATION OF NETWORK FUNCTIONS

One of the initial steps undertaken in this study was the selection of a computer program that generates network functions. The program chosen was given by Chang [3]. This program, written in the FORTRAN IV language, is based on topological formulas and uses the "dichotomy method" of a tree finding to generate the network functions. The program was modified to have it generate only the voltage transfer functions.

The program to be presented as a result of this investigation did not substantially alter the tree-finding sections of Chang's program. Therefore, a discussion of the theory and operation of the program taken from [3] will be given only when it is pertinent to the contents of this paper.

B. PROGRAM LIMITATIONS

Some limitations of the program, which will be presented, are now given. The program limits the networks that will be admitted to those with graphs having the following properties: (a) non-oriented, (b) linear, (c) a maximum of 15 nodes, and (d) a maximum of 60 edges [3].

The network functions are not generated in symbolic form (i.e., the program requires that the input data, the network elements,

have numerical values). This requires that the tree-finding subroutine be repeated each time that an element is varied, during the process of obtaining the data necessary for fault identification. The result of this is a substantial increase in the computer run time.

These limitations do not impair the validity of this investigation, since it is in the nature of a feasibility study.

Additional comments on this subject will be made in a following section.

C. CATEGORY OF NETWORKS CONSIDERED

The limitations given in the preceding section place a restriction on the class of networks that will be allowed. The networks considered will be passive, linear networks (i. e., containing only linear resistive, inductive, and capacitive elements). However, in this class of networks, the purely resistive networks will be prohibited, because of the frequency-dependent nature that is inherent in the proposed test procedure.

The assumption is also made that the networks under consideration have a defined set of input and output terminals. Thus, the permitted class of networks is: the set of passive linear, two-port networks.

An illustration of a general two-port network is shown in Figure 1. The input and output terminals are represented by (1, 3) and (2, 4) respectively.

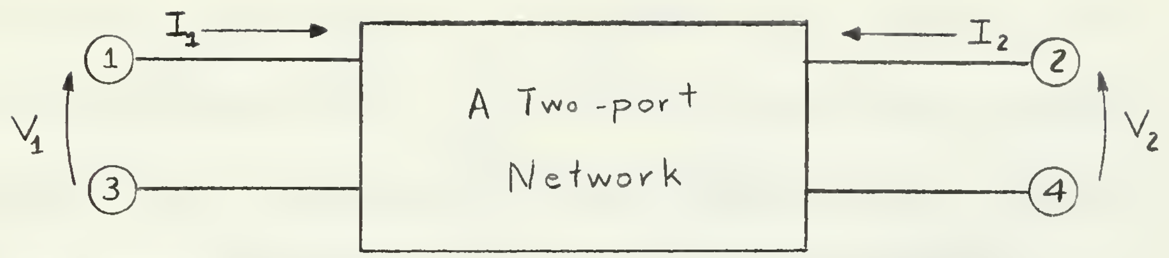

Figure 1

The properties of two-port networks are well covered in the literature [3] - [6]. The only two-port properties that are necessary for this investigation are the forward and reverse voltage-transfer functions. These are defined by equations (1) and (2) respectively.

$$G_{12} = \left. \frac{V_2}{V_1} \right|_{I_2 = 0} \tag{1}$$

$$G_{21} = \left. \frac{V_1}{V_2} \right|_{I_1 = 0} \tag{2}$$

D. VOLTAGE TRANSFER FUNCTIONS

In the preceding section, it was stated that the only two-port properties needed were the voltage transfer functions. A number of factors were considered before this decision was reached.

First, the selection of the voltage transfer functions leads to a more simplified and convenient test procedure than would be obtained if some of the other network functions were selected. The test procedure proposed under this method will require only a

signal generator and a voltmeter. Obviously, using a combination of the other network functions, such as the z-parameters or current gains, would require more test equipment. Also, in the event of an emergency equipment failure, the test instruments most likely to be available would be a VTVM and a signal generator. This would be the only test equipment necessary in a program of preventive maintenance, which is the desired goal to be obtained from an extension of the test procedure presented here. The logical conclusion reached from these points is to use the voltage transfer functions.

In the preceding discussion, the suggestion was made to use the voltage transfer functions in the test procedure. The choice must then be made to use one or both of the voltage transfer functions. The decision was made to use both of these transfer functions. The factors leading to this decision are now given.

In general, a two-port network would have one set of terminals for the normal input and the other set for the normal output. In this situation, the reverse voltage-transfer function may be considered to be of very minor importance. However, it may provide data that is useful or even necessary to identify a faulty component.

For any given network, the possibility exists that an element or elements of that network may not appear in the expression of one or more of the network functions. As an example, consider the ladder network of Figure 2.

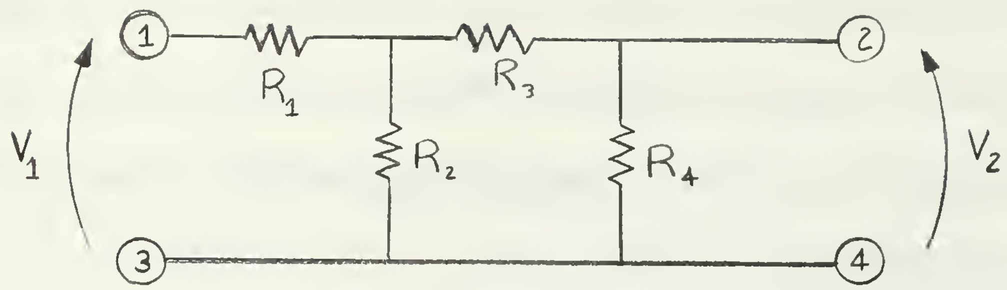

Figure 2

The voltage transfer functions, as defined by equations (1) and (2), can be found by a number of methods [3], [4]. These have been calculated and are given in equations (3) and (4).

$$G_{12} = \frac{R_2 R_4}{R_1(R_2 + R_3 + R_4) + R_2(R_3 + R_4)} \tag{3}$$

$$G_{21} = \frac{R_2}{R_2 + R_3} \tag{4}$$

An inspection of these equations shows that elements R_1 and R_4 do not appear in the equation for G_{21}. For the proposed fault-identification procedure this means, of course, that no information can be obtained about the values of R_1 and R_4 by making tests on G_{21}.

Another reason for selecting both of the voltage transfer functions is the possibility that a change in the value of a network element may be offset by a change in one or more other elements, with the result that the magnitude of the transfer function will remain unchanged.

This can be illustrated by again considering the network of Figure 2. Assuming that all of the element values are equal to 1, equations (5) and (6) are obtained by substituting these values into (3) and (4).

$$G_{12} = \frac{1}{5} \tag{5}$$

$$G_{21} = \frac{1}{2} \tag{6}$$

Now assume that R_2 has increased in value to 3, that R_3 has increased in value to 2, and that R_1 and R_4 have remained unchanged. The voltage transfer functions can again be evaluated by substituting these new network values into equations (3) and (4). The results of this are given in equations (7) and (8).

$$G_{12} = \frac{1}{5} \tag{7}$$

$$G_{21} = \frac{3}{5} \tag{8}$$

A comparison of equations (5) and (7) shows that the magnitude of G_{12} has remained unchanged, although one of the network elements has been increased in value by 300% and another by 200%. However, when equations (6) and (8) are compared, the change in magnitude of G_{21} indicates that a change has occurred in at least one of the network elements R_2 or R_3.

It is noted that no useful information for fault identification purposes would be obtained by making measurements of the voltage

transfer functions in the example given above, and it was stated earlier that purely resistive networks would not be allowed. However, this argument can readily be extended to cover the case of a network for which the magnitudes of its transfer functions are frequency dependent.

One other factor will be noted that argues for the use of both of the voltage transfer functions. It is, of course, the general catch-all phrase of having non-ideal test equipment and operators. In general, the test equipment should be operating within its performance specifications and the operator experienced enough in the use of this equipment so that the probability of an error in measurement would be lessened due to the larger number of measurements required from the use of both transfer functions.

A measure of flexibility is added to the procedure through the use of both the forward and reverse voltage-transfer functions. The data for fault identification purposes will be in the form of two tables, one for each transfer function. It is possible that the information needed can be determined by making measurements on only one of the transfer functions and the corresponding table for that transfer function. If the needed information cannot be determined from this one set, the other would be available. It could also be used to serve as a check on the first set of measurements made. A form of this flexibility can be seen in the following example.

For the network of Figure 2, assume that R_1 has been removed. It is obvious that G_{12} will be equal to 0, and in no way will it be affected by the remaining network elements. However, a test made on G_{12} will provide data on the values of R_2 and R_3.

III. TEST FREQUENCIES

The requirement that the response of a network, the voltage transfer functions in particular, be frequency dependent was given in Chapter II. A discussion of some of the applicable properties, in the frequency domain, of network functions is given, and the choice of test frequencies is discussed.

A. GENERAL PROPERTIES

The frequency-domain properties of network functions are covered in the literature [4], [7]. Some of the properties pertinent to this investigation are now given.

The frequency-domain representation of a network function for a network consisting of linear, time-invariant R, L, and C elements can be given in the form of equation (9).

$$H(s) = \frac{a_0 + a_1 s + \cdots + a_n s^n}{s^k \left(b_0 + b_1 s + \cdots + b_m s^m\right)} \tag{9}$$

The function H(s) is a general network function, where s is the complex variable, frequency. The a and b coefficients are required to be real, non-negative, and constant. The relationships between k, m, and n depend upon the particular network function and the elements making up the network.

Although the constraints on the coefficients of equation (9) are important, the polynomial form of this equation is not useful

for this investigation. Therefore, equation (9) can be factored and given in the form of equation (10).

$$H(s) = K \frac{(s-z_1)(s-z_2)\cdots(s-z_n)}{s^k(s-p_1)(s-p_2)\cdots(s-p_m)} \tag{10}$$

The constant K is a gain term that is determined by the coefficients a_n and b_m of equation (9). The z and p terms are defined as the zeros and poles of the function. Although the coefficients of equation (9) are real, the poles and zeros of the function may be imaginary or complex.

When imaginary or complex roots do occur, it is always in the form of conjugate pairs. This means that any imaginary or complex terms appearing in (10) may be eliminated by multiplying together the appropriate roots. When this is done, the network function is given in the form of equation (11).

$$H(s) = K \frac{(s-z_1)(s-z_2)\cdots(s-z_i)\cdots(s^2+\alpha s+\beta)}{s^k(s-p_1)(s-p_2)\cdots(s-p_j)\cdots(s^2+\delta s+\epsilon)} \tag{11}$$

Equation (11) is the form that is most useful for this study. The coefficients of all of the factors appearing in this equation are real.

B. CORNER FREQUENCIES

In this study the interest is in the sinusoidal steady-state operation of a network (i.e., $s = j\omega$). Therefore, a frequency-response approach is used to investigate the factors making up

equation (11). Only the first-order and second-order factors are discussed, since they are the primary test frequencies.

1. <u>First-Order Factors</u>

Consider the first-order factor $(s-z_i)$ in equation (11). For sinusoidal steady-state operation, this factor can be re-defined by equation (12).

$$s - z_i \equiv j\omega + \omega_c \tag{12}$$

As an example, consider the voltage transfer function given in equation (13). In order to use the frequency-response approach, the magnitude of this function can be expressed in dB and given by equation (14).

$$G_{12}(j\omega) = j\omega + \omega_c \tag{13}$$

$$g_{12} \equiv 20 \log_{10} \left| G_{12}(j\omega) \right| = 20 \log_{10} \left| j\omega + \omega_c \right| = 20 \log_{10} \left| 1 + j\frac{\omega}{\omega_c} \right| \tag{14}$$

At extremely low frequencies ($\omega \ll \omega_c$), (14) can be approximated by (15).

$$g_{12} \approx 20 \log_{10} |1| = 0 \tag{15}$$

For the extremely high frequencies ($\omega \gg \omega_c$), the approximation in (16) can be obtained.

$$g_{12} \approx 20 \log_{10} \left| j\frac{\omega}{\omega_c} \right| \tag{16}$$

The straight-line asymptotic approximation to (14) is given by (15) and (16). (15) is for the frequencies when $\omega < \omega_c$, and (16) is for frequencies where $\omega > \omega_c$. A plot of this approximation is given in Figure 3, where the frequency axis is normalized with respect to ω_c.

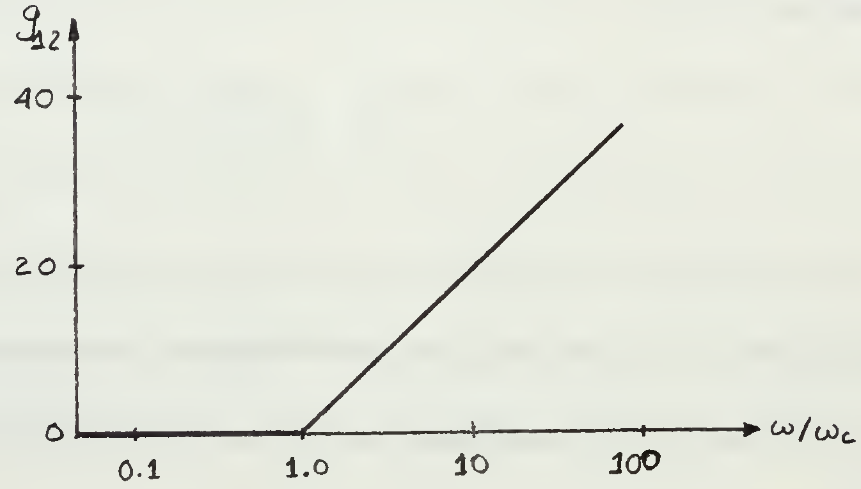

Figure 3

The actual value of the transfer function at the frequency where $\omega = \omega_c$ is 3 dB above the value of the straight-line approximation of Figure 3 (it would be 3 dB below the value of the straight-line approximation if the network function had consisted of a pole). The slope of the approximation is 0 for $\omega < \omega_c$ and 20 dB per decade of frequency for $\omega > \omega_c$ (-20 dB per decade of frequency when the factor is a pole).

The frequency ω_c is defined as the corner frequency of a first-order factor. The corner frequencies for first-order factors are selected as some of the test frequencies.

2. Second-Order Factors

The second-order or quadratic factors of equation (11) are considered. Factors of this form can be re-defined as in equation (17).

$$S^2 + \alpha S + \beta \equiv (j\omega)^2 + 2\zeta\omega_n(j\omega) + \omega_n^2 \tag{17}$$

ω_n is defined as the undamped natural frequency, and ζ is the damping factor.

The quadratic factor can be investigated in the same manner that was used for the first-order factor. The transfer function considered for this factor is given in equation (18).

$$G_{12}(j\omega) = (j\omega)^2 + 2\zeta\omega_n(j\omega) + \omega_n^2 \tag{18}$$

Without carrying out the steps for the investigation of the first-order factor, the results of the analysis on the quadratic factor are given in Figure 4 [7].

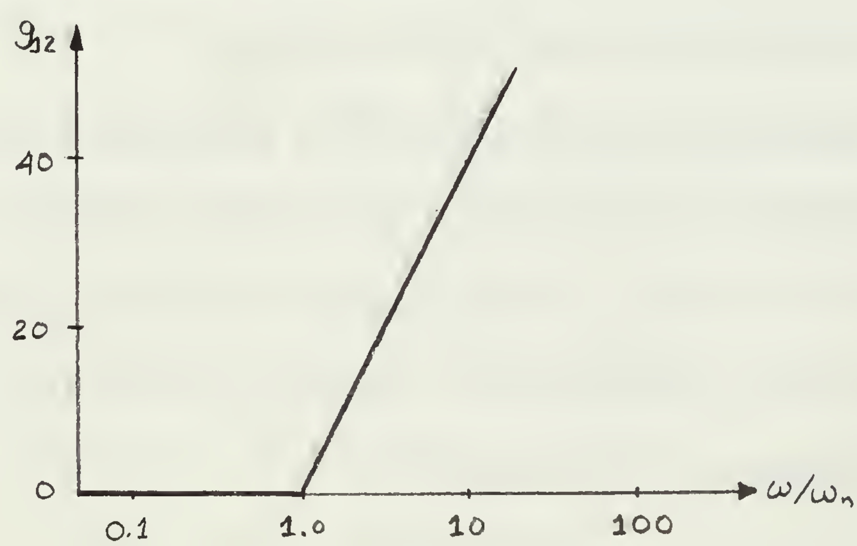

Figure 4

The slope of the curve in Figure 4 for $\omega < \omega_n$ is 0; the slope is 40 dB per decade of frequency for $\omega > \omega_n$. The frequency axis has been normalized with respect to ω_n.

The frequency ω_n is defined as the corner frequency for a quadratic factor. Frequencies resulting from this type of factors are used as test frequencies.

Some comments on quadratic factors are now given. The quadratic factor has a peak value when the damping ratio is less than $(\sqrt{2})^{-1}$. The expressions for the magnitude of the peak value and the frequency at which it occurs are known [7]. These are dependent upon the value of the damping ratio. Although these are important quantities and the peak value of the function can be large compared with the value of the straight-line approximation at that frequency, these have not been considered in this study. When this condition occurs in a complex system, a simple analytic expression relating the corner frequency and the frequency at which the peak value occurs may not be available.

C. ADDITIONAL TEST FREQUENCIES

Although the poles and zeros of a function uniquely determine that function (to within an all-pass function) [1], the corner frequencies are not adequate in themselves to determine the necessary data for fault identification. Therefore, it became necessary to select additional test frequencies.

The selection of these additional test frequencies was done in a simple and arbitrary manner. However, very encouraging results have been obtained from the sample problems that have been worked using this method.

The selection of these additional test frequencies was done in subroutine RESULT of the computer program, and this method is now described.

The corner frequencies were calculated and placed into array WA. The array was then sorted numerically by eliminating zero frequencies and placing the first non-zero frequency into the first position in the array. The zero frequencies were eliminated because of the sinusoidal steady-state operation being considered. The non-zero frequencies were compared, and when identical frequencies occurred, all but one were eliminated.

Two additional frequencies were selected as follows: the lowest non-zero corner frequency was divided by ten, and the highest corner frequency was multiplied by two.

The other frequencies were selected by comparing the ratio of adjacent corner frequencies. When this ratio was less than two, no additional frequency was chosen. When the ratio was less than ten, one additional frequency was selected by taking the arithmetic mean of the corner frequencies. When the ratio was less than 100, two additional frequencies were chosen as follows: one was twice the value of the lower corner frequency and the other was half the

value of the higher corner frequency. When the ratio was greater than 100, three additional frequencies were selected as follows: one was ten times the value of the lower corner frequency, one was one-tenth the value of the higher corner frequency, and the third was the arithmetic mean of the corner frequencies.

All of these additional frequencies plus the corner frequencies were placed into array WB. This array was then sorted in the same manner as WA. The ratios of the adjacent frequencies in array WB were compared, and the higher of the two frequencies were eliminated if this ratio was found to be less than 1.01. The frequencies then remaining in array WB constituted the set of test frequencies.

D. OTHER POSSIBLE TEST FREQUENCIES

The selection of the additional test frequencies, as given in the previous section, was completely arbitrary. Although the results obtained using this method have been very good, another completely arbitrary method could have been used. Perhaps fewer test frequencies would be required, and the results obtained might be as good or better.

Another possible source of test frequencies would be to investigate the frequency response of the function to obtain the frequencies at which relative extrema occur [8]. At frequencies near these extrema, the slope of the curve should not be changing very rapidly. This would mean that an error in the setting of the frequency of a signal generator would not be as probable to give misleading information.

IV. FAULT ISOLATION SIGNATURES

A. GENERAL

Once the test frequencies have been selected, the next step is to use these frequencies to obtain the necessary data for fault identification purposes.

The magnitudes of the voltage transfer functions are calculated for the nominal network values at each of the test frequencies. These magnitude ratios are then expressed in dB by multiplying 20 times the common logarithm of the magnitude ratio.

Next, the network elements have to be varied in value. This can be done over whatever range and in any combinations that are desired. The transfer functions are calculated for each variation and evaluated at the same set of test frequencies (also expressed in dB).

The data obtained from these steps can be handled in any way that provides desired information.

B. A SET OF SIGNATURES

The evaluations of the transfer functions expressed in dB (or as a simple ratio) as given above would not be extremely useful in this form. For example, if this data were put into the form of a table to be used for comparison purposes, the table would be lengthy and difficult to read. Also, it is doubtful that an exact match would

be obtained when measurements were made on an actual network with the entries of the table.

Therefore, it was decided to associate a single integer number (signature) for each value of the transfer functions at each of the test frequencies. This results in an n by m table where m is the number of test frequencies and n is the number of variations made on the network elements, and each entry in the table is a single integer number.

The signatures were determined in the following manner. First, the gains of the functions for the varied elements were compared to the gains of the nominal network at the test frequencies. This was done by taking the absolute value of the difference between the nominal and varied gains. Next, signatures were assigned according to the format of Table I.

Absolute value of difference in gain (dB)	Signature assigned
less than 0.5	0
0.5 to 1.0	1
1.0 to 1.5	2
1.5 to 2.0	3
2.0 to 3.0	4
3.0 to 4.0	5
4.0 to 5.0	6
5.0 to 6.0	7
6.0 to 8.0	8
greater than 8.0	9

Table I

The assigning of these signatures has been arbitrary. If only a specific type of network were to be studied, a specific set of signatures could be devised for this particular problem. At least one different, but similar method to the one given here has been used [1]. In this method gains above and below the nominal value were considered separately, and the signatures were assigned accordingly.

An "arbitrary" method of assigning signatures may be necessary, because different types of networks may be assigned signatures from the same program. However, for a complex system it may be necessary to develop a specific set of signatures.

V. RESULTS OBTAINED

A. DISCUSSION OF THE PROGRAM

A brief discussion of the program that is presented is now given. As given in Chapter II, the program taken from [3] was modified to generate only the voltage transfer functions. The only major changes made to this program were in MAIN and subroutine RESULT. Therefore, a discussion of the operation of the remaining subroutines is not given.

Essentially the operation of the program up to the point where RESULT is called was not modified. When RESULT is called, the transfer functions for the nominal network are calculated in polynomial form. These are printed as output data. Then, the IBM supplied subroutine POLRT is called to evaluate the poles and zeros of the transfer functions. These roots are used to determine the set of test frequencies which are stored in array WC. The transfer functions for the nominal network are then evaluated at the test frequencies. The test frequencies (in Hz) and the corresponding values of the transfer functions (in dB) are printed as output data. The computation is then returned to MAIN.

In MAIN a counter, IA, is updated and one of the network elements is varied in value. The tree-finding portions of the program are again repeated until RESULT is called. In RESULT the transfer functions for the modified network are calculated in polynomial form.

These are evaluated at the test frequencies and signatures assigned by comparing these values with those for the nominal network. The following is then printed as output: the element varied, its deviation from the nominal value, and the associated signatures. This process is repeated for each variation of an element.

The elements were varied near the end of MAIN, and only one element was varied at a time. By placing nested DO loops after statement number 210 in MAIN, the elements can be varied in whatever combination is desired. However, this was not done due to the excessive amount of printed output and computing time that would have been required.

B. EXAMPLES

Two examples have been included that are typical of the results that have been obtained. These results are given in the COMPUTER OUTPUT section.

1. Example 1

The network used for Example 1 is shown in Figure 5.

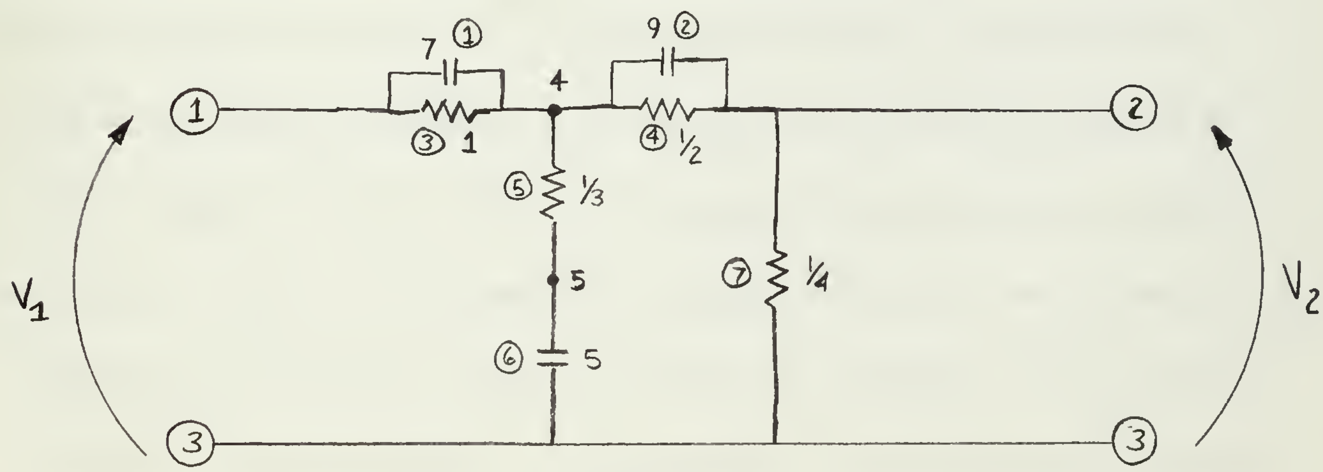

Figure 5

In Figure 5 the nominal element values have been labelled. The element number has been given in a circle by its corresponding element, and the node numbers are given as required in [3].

2. Example 2

The network used for Example 2 is given in Figure 6. The same discussion given in Example 1 applies here.

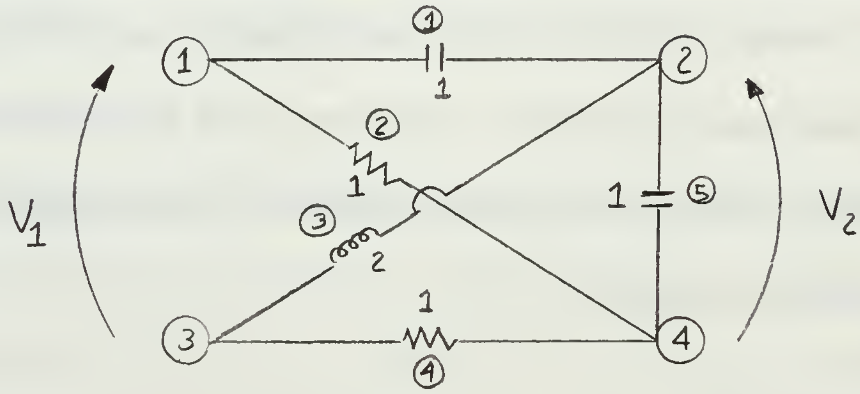

Figure 6

The results obtained from these examples and others have been very encouraging. It can be seen from the COMPUTER OUTPUT section that the varied element can be easily identified.

VI. CONCLUSIONS

The results obtained from this study indicate that the method presented here should be of practical value in a maintenance program (either preventive or corrective). The method of selecting test frequencies has been shown to be a sound one. The assignment of signatures can be made flexible to meet various requirements by having a few interchangeable sets of signatures to be entered into the computer program.

The computer program is not practical in its present form. It should be changed to generate the transfer functions in symbolic form. This is necessary to insure a reasonable computing time when the large number of combinations for the variations are considered, because the tree-finding portions of the program could then be bypassed. Also, modifications need to be made to allow for the inclusion of mutual inductances and active network elements.

The output of the program was left in a simple form because of the variety of ways in which it can be used. For example, the set of signatures (with the associated elements) could be sorted into some logical sequence to be placed into a table that would be used to look up faulty elements from actual measurements made on a circuit. Also, by expanding the program to allow the measurements made on a circuit to be used as inputs, the computer could compare these with its calculated set of signatures to provide only the faulty elements as output.

INPUT DATA

NODES= 5 EDGES= 7 GROUND CONDITION= 1

EDGE NUMBER	NODE NUMBERS		ADMITTANCE	POWER
1	1	4	7.0000	1
2	2	4	9.0000	1
3	1	4	1.0000	0
4	2	4	2.0000	0
5	4	5	3.0000	0
6	3	5	5.0000	1
7	2	3	4.0000	0

A MATRIX

1	0	1	0	0	0	0
0	1	0	1	0	0	1
0	0	0	0	0	1	1
1	1	1	1	1	0	0
0	0	0	0	1	1	0

```
                6.00S** 0+ 79.00S** 1+304.00S** 2+315.00S** 3
V2/V1 = ------------------------------------------------------------------------------------
               42.00S** 0+421.00S** 1+759.00S** 2+315.00S** 3
```

TEST FREQUENCY (HZ)	GAIN (DB)
0.203E-02	-0.169E 02
0.203E-01	-0.161E 02
0.227E-01	-0.159E 02
0.354E-01	-0.146E 02
0.654E-01	-0.114E 02
0.955E-01	-0.896E 01
0.101E 00	-0.858E 01
0.182E 00	-0.483E 01
0.262E 00	-0.299E 01
0.524E 00	-0.964E 00

```
                6.00S** 0+ 79.00S** 1+304.00S** 2+315.00S** 3
V1/V2 = ------------------------------------------------------------------------------------
                6.00S** 0+ 94.00S** 1+409.00S** 2+315.00S** 3
```

TEST FREQUENCY (HZ)	GAIN (DB)
0.207E-02	-0.270E-01
0.207E-01	-0.160E 01
0.227E-01	-0.178E 01
0.354E-01	-0.257E 01
0.654E-01	-0.293E 01
0.955E-01	-0.257E 01
0.163E 00	-0.160E 01
0.326E 00	-0.579E 00

FAULT ISOLATION SIGNATURES FOR V2/V1

ELEMENT	DEVIATION	SIGNATURES									
1	-0.9	0	2	3	5	8	9	9	9	9	9
1	-0.7	0	2	3	4	7	8	8	8	7	5
1	-0.5	0	2	2	4	5	5	5	5	4	2
1	-0.3	0	1	1	2	3	3	3	3	2	1
1	-0.1	0	0	0	0	1	1	1	0	0	0
1	0.1	0	0	0	0	0	0	0	0	0	0
1	0.3	0	1	1	2	2	2	2	1	1	0
1	0.5	0	2	2	3	3	3	3	2	1	0
1	0.7	0	3	3	4	4	4	4	3	2	0
1	0.9	0	3	4	4	4	4	4	3	2	0
1	1.1	0	4	4	4	5	4	4	4	2	1
1	1.3	0	4	4	5	5	5	5	4	3	1
1	1.5	0	4	5	5	5	5	5	4	3	1
2	-0.9	0	0	0	2	5	6	6	8	8	7
2	-0.7	0	0	0	2	4	5	5	5	4	2
2	-0.5	0	0	0	1	3	4	4	3	2	1
2	-0.3	0	0	0	1	2	2	2	1	1	0
2	-0.1	0	0	0	0	0	0	0	0	0	0
2	0.1	0	0	0	0	0	0	0	0	0	0
2	0.3	0	0	0	1	1	1	1	0	0	0
2	0.5	0	0	1	1	2	2	1	1	0	0
2	0.7	0	1	1	2	2	2	2	1	0	0
2	0.9	0	1	1	2	3	2	2	1	1	0
2	1.1	0	1	2	3	3	3	3	1	1	0
2	1.3	0	2	2	3	4	3	3	1	1	0
2	1.5	0	2	2	4	4	3	3	2	1	0
3	-0.9	9	3	2	0	0	0	0	0	0	0
3	-0.7	8	3	2	0	0	0	0	0	0	0
3	-0.5	5	2	2	0	0	0	0	0	0	0
3	-0.3	3	1	1	0	0	0	0	0	0	0
3	-0.1	1	0	0	0	0	0	0	0	0	0
3	0.1	0	0	0	0	0	0	0	0	0	0
3	0.3	2	1	1	0	0	0	0	0	0	0
3	0.5	3	2	2	1	0	0	0	0	0	0
3	0.7	4	3	3	2	0	0	0	0	0	0
3	0.9	4	4	3	2	0	0	0	0	0	0
3	1.1	5	4	4	3	1	0	0	0	0	0
3	1.3	5	4	4	3	1	0	0	0	0	0
3	1.5	5	5	4	4	1	0	0	0	0	0

ELEMENT	DEVIATICN	SIGNATURES									
4	-0.9	9	4	3	1	0	0	0	0	0	0
4	-0.7	6	4	3	1	0	0	0	0	0	0
4	-0.5	4	3	2	1	0	0	0	0	0	0
4	-0.3	2	1	1	0	0	0	0	0	0	0
4	-0.1	0	0	0	0	0	0	0	0	0	0
4	0.1	0	0	0	0	0	0	0	0	0	0
4	0.3	1	1	1	0	0	0	0	0	0	0
4	0.5	1	2	2	1	0	0	0	0	0	0
4	0.7	2	2	2	2	0	0	0	0	0	0
4	0.9	2	3	3	2	0	0	0	0	0	0
4	1.1	2	3	3	3	1	0	0	0	0	0
4	1.3	3	3	3	3	1	0	0	0	0	0
4	1.5	3	4	4	3	1	0	0	0	0	0
5	-0.9	0	0	1	2	4	4	4	3	2	0
5	-0.7	0	0	0	1	2	3	3	2	1	0
5	-0.5	0	0	0	0	1	1	1	1	1	0
5	-0.3	0	0	0	0	0	0	1	1	0	0
5	-0.1	0	0	0	0	0	0	0	0	0	0
5	0.1	0	0	0	0	0	0	0	0	0	0
5	0.3	0	0	0	0	0	0	0	0	0	0
5	0.5	0	0	0	0	0	0	0	1	1	0
5	0.7	0	0	0	0	0	0	1	1	1	0
5	0.9	0	0	0	0	0	1	1	2	2	1
5	1.1	0	0	0	0	0	1	1	2	2	1
5	1.3	0	0	0	0	0	1	1	2	2	1
5	1.5	0	0	0	0	0	1	1	2	3	2
6	-0.9	0	2	2	3	4	4	4	2	1	0
6	-0.7	0	1	2	2	3	2	2	1	0	0
6	-0.5	0	1	1	2	2	1	1	0	0	0
6	-0.3	0	0	0	1	1	0	0	0	0	0
6	-0.1	0	0	0	0	0	0	0	0	0	0
6	0.1	0	0	0	0	0	0	0	0	0	0
6	0.3	0	0	0	1	0	0	0	0	0	0
6	0.5	0	1	1	1	1	0	0	0	0	0
6	0.7	0	1	2	2	1	0	0	0	0	0
6	0.9	0	2	2	2	1	0	0	0	0	0
6	1.1	0	2	3	3	1	0	0	0	0	0
6	1.3	0	3	3	3	1	0	0	0	0	0
6	1.5	0	3	4	3	1	0	0	0	0	0
7	-0.9	9	9	9	9	8	7	7	4	3	1
7	-0.7	8	8	8	8	7	6	5	4	3	1
7	-0.5	6	6	6	6	5	4	4	3	2	0
7	-0.3	4	4	4	4	3	3	3	2	1	0
7	-0.1	1	1	1	1	1	1	1	0	0	0
7	0.1	1	1	1	1	1	1	1	0	0	0
7	0.3	3	3	3	3	3	2	2	2	1	0
7	0.5	5	4	4	4	4	4	4	3	2	1
7	0.7	6	5	5	5	5	5	5	4	3	1
7	0.9	6	6	6	6	6	5	5	4	4	2
7	1.1	7	7	7	7	7	6	6	5	4	2
7	1.3	8	8	8	8	7	7	7	6	5	3
7	1.5	8	8	8	8	8	7	7	6	5	3

FAULT ISOLATION SIGNATURES FOR V1/V2

ELEMENT	DEVIATICN	SIGNATURES							
1	-0.9	0	0	0	0	0	0	0	0
1	-0.7	0	0	0	0	0	0	0	0
1	-0.5	0	0	0	0	0	0	0	0
1	-0.3	0	0	0	0	0	0	0	0
1	-0.1	0	0	0	0	0	0	0	0
1	0.1	0	0	0	0	0	0	0	0
1	0.3	0	0	0	0	0	0	0	0
1	0.5	0	0	0	0	0	0	0	0
1	0.7	0	0	0	0	0	0	0	0
1	0.9	0	0	0	0	0	0	0	0
1	1.1	0	0	0	0	0	0	0	0
1	1.3	0	0	0	0	0	0	0	0
1	1.5	0	0	0	0	0	0	0	0
2	-0.9	0	1	1	0	3	5	7	7
2	-0.7	0	0	0	0	3	4	5	4
2	-0.5	0	0	0	0	2	3	3	1
2	-0.3	0	0	0	0	1	1	1	0
2	-0.1	0	0	0	0	0	0	0	0
2	0.1	0	0	0	0	0	0	0	0
2	0.3	0	0	0	0	1	1	0	0
2	0.5	0	0	0	0	1	1	1	0
2	0.7	0	0	0	1	2	2	1	0
2	0.9	0	0	0	1	2	2	1	0
2	1.1	0	0	0	1	2	2	1	0
2	1.3	0	0	0	1	2	2	1	0
2	1.5	0	0	0	2	3	3	2	0
3	-0.9	0	0	0	0	0	0	0	0
3	-0.7	0	0	0	0	0	0	0	0
3	-0.5	0	0	0	0	0	0	0	0
3	-0.3	0	0	0	0	0	0	0	0
3	-0.1	0	0	0	0	0	0	0	0
3	0.1	0	0	0	0	0	0	0	0
3	0.3	0	0	0	0	0	0	0	0
3	0.5	0	0	0	0	0	0	0	0
3	0.7	0	0	0	0	0	0	0	0
3	0.9	0	0	0	0	0	0	0	0
3	1.1	0	0	0	0	0	0	0	0
3	1.3	0	0	0	0	0	0	0	0
3	1.5	0	0	0	0	0	0	0	0

ELEMENT	DEVIATION	SIGNATURES							
4	-0.9	2	4	3	1	0	0	0	0
4	-0.7	0	3	3	1	0	0	0	0
4	-0.5	0	2	2	1	0	0	0	0
4	-0.3	0	1	1	0	0	0	0	0
4	-0.1	0	0	0	0	0	0	0	0
4	0.1	0	0	0	0	0	0	0	0
4	0.3	0	0	0	0	0	0	0	0
4	0.5	0	1	1	1	0	0	0	0
4	0.7	0	1	1	1	0	0	0	0
4	0.9	0	1	1	2	1	0	0	0
4	1.1	0	1	2	2	1	0	0	0
4	1.3	0	2	2	2	1	0	0	0
4	1.5	0	2	2	2	1	0	0	0
5	-0.9	0	1	1	3	4	4	3	1
5	-0.7	0	0	0	1	3	3	2	0
5	-0.5	0	0	0	0	1	2	1	0
5	-0.3	0	0	0	0	0	1	1	0
5	-0.1	0	0	0	0	0	0	0	0
5	0.1	0	0	0	0	0	0	0	0
5	0.3	0	0	0	0	0	0	0	0
5	0.5	0	0	0	0	0	1	1	0
5	0.7	0	0	0	0	0	1	1	1
5	0.9	0	0	0	0	0	1	2	1
5	1.1	0	0	0	0	0	1	2	1
5	1.3	0	0	0	0	0	1	2	2
5	1.5	0	0	0	0	0	1	2	2
6	-0.9	0	2	3	4	4	4	2	0
6	-0.7	0	2	2	3	3	2	0	0
6	-0.5	0	1	1	2	2	1	0	0
6	-0.3	0	1	1	1	1	0	0	0
6	-0.1	0	0	0	0	0	0	0	0
6	0.1	0	0	0	0	0	0	0	0
6	0.3	0	1	1	1	0	0	0	0
6	0.5	0	1	1	1	1	0	0	0
6	0.7	0	2	2	2	1	0	0	0
6	0.9	0	2	3	3	1	0	0	0
6	1.1	0	3	3	3	1	0	0	0
6	1.3	0	4	4	3	1	0	0	0
6	1.5	0	4	4	4	1	0	0	0
7	-0.9	0	0	0	0	0	0	0	0
7	-0.7	0	0	0	0	0	0	0	0
7	-0.5	0	0	0	0	0	0	0	0
7	-0.3	0	0	0	0	0	0	0	0
7	-0.1	0	0	0	0	0	0	0	0
7	0.1	0	0	0	0	0	0	0	0
7	0.3	0	0	0	0	0	0	0	0
7	0.5	0	0	0	0	0	0	0	0
7	0.7	0	0	0	0	0	0	0	0
7	0.9	0	0	0	0	0	0	0	0
7	1.1	0	0	0	0	0	0	0	0
7	1.3	0	0	0	0	0	0	0	0
7	1.5	0	0	0	0	0	0	0	0

TIME= 30.31 SEC

INPUT DATA

NODES= 4   EDGES= 5   GROUND CONDITION= 2

EDGE NUMBER	NODE NUMBERS		ADMITTANCE	PCWER
1	1	2	1.0000	1
2	1	4	1.0000	0
3	2	3	2.0000	-1
4	3	4	1.0000	0
5	2	4	1.0000	1

A MATRIX

```
1  1  C  0  0
1  0  1  0  1
0  0  1  1  0
0  1  C  1  1
```

VCLTAGE TRANSFER FUNCTIONS

```
              -2.00S** 0+  1.00S** 2
V2/V1 = -------------------------------------------------------------------
               4.00S** 0+  2.00S** 1+  4.00S** 2+  1.00S** 3
```

TEST FREQUENCY (HZ)	GAIN (DB)
0.164E-01	-0.589E 01
0.164E 00	0.973E 01
0.225E 00	0.249E-04
0.411E 00	-0.943E 01
0.597E 00	-0.127E 02
0.119E 01	-0.180E 02

```
              -2.00S** 0+  1.00S** 2
V1/V2 = -------------------------------------------------------------------
               2.00S** 0+  3.00S** 1+  1.00S** 2
```

TEST FREQUENCY (HZ)	GAIN (DB)
0.159E-01	-0.107E-01
0.159E 00	-0.458E 00
0.225E 00	-0.512E 00
0.318E 00	-0.458E 00
0.637E 00	-0.209E 00

FAULT ISOLATION SIGNATURES FOR V2/V1

ELEMENT	DEVIATICN	SIGNATURES					
1	-0.9	0	9	2	5	7	8
1	-0.7	0	9	2	2	3	3
1	-0.5	0	8	3	0	1	1
1	-0.3	0	5	2	0	0	0
1	-0.1	0	1	0	0	0	0
1	0.1	0	0	0	0	0	0
1	0.3	0	0	2	0	0	0
1	0.5	0	2	3	0	0	0
1	0.7	0	5	4	0	0	0
1	0.9	0	6	4	0	0	0
1	1.1	0	7	5	0	0	0
1	1.3	0	8	5	0	0	0
1	1.5	0	8	5	0	0	0
2	-0.9	9	8	0	2	2	0
2	-0.7	8	7	0	2	1	0
2	-0.5	5	5	0	1	1	0
2	-0.3	3	3	0	0	0	0
2	-0.1	0	1	0	0	0	0
2	0.1	0	1	0	0	0	0
2	0.3	2	2	0	0	0	0
2	0.5	3	4	0	1	1	0
2	0.7	3	5	0	1	1	1
2	0.9	4	5	0	2	2	1
2	1.1	4	6	0	2	2	1
2	1.3	4	6	0	2	3	2
2	1.5	5	7	0	3	3	2
3	-0.9	2	9	9	5	3	1
3	-0.7	0	9	9	4	2	0
3	-0.5	0	9	8	4	2	0
3	-0.3	0	7	5	2	1	0
3	-0.1	0	1	2	0	0	0
3	0.1	0	1	2	0	0	0
3	0.3	0	5	5	2	1	0
3	0.5	0	7	8	4	2	0
3	0.7	0	8	8	4	2	0
3	0.9	0	9	8	5	3	1
3	1.1	0	9	7	6	4	1
3	1.3	0	9	6	7	4	1
3	1.5	0	9	5	8	5	1

ELEMENT	DEVIATION	SIGNATURES					
4	-0.9	7	5	0	8	9	9
4	-0.7	5	4	0	6	8	9
4	-0.5	4	3	0	4	5	7
4	-0.3	2	1	0	2	3	4
4	-0.1	0	0	0	0	1	1
4	0.1	0	0	0	0	0	1
4	0.3	2	1	0	1	2	3
4	0.5	3	2	0	2	3	4
4	0.7	4	3	0	3	4	5
4	0.9	5	4	0	3	4	6
4	1.1	5	4	0	4	5	7
4	1.3	6	4	0	4	5	7
4	1.5	6	5	0	4	5	8
5	-0.9	0	6	9	8	9	9
5	-0.7	0	4	9	7	7	8
5	-0.5	0	1	8	5	5	6
5	-0.3	0	4	5	3	4	4
5	-0.1	0	2	1	1	1	1
5	0.1	0	2	1	1	1	1
5	0.3	0	5	4	3	3	4
5	0.5	0	7	5	4	4	5
5	0.7	0	8	6	5	5	6
5	0.9	0	9	7	6	6	7
5	1.1	0	9	8	6	7	7
5	1.3	0	9	8	7	7	8
5	1.5	0	9	8	8	8	8

FAULT ISOLATION SIGNATURES FOR V1/V2

ELEMENT	DEVIATION	SIGNATURES				
1	-0.9	0	0	1	2	4
1	-0.7	0	0	0	0	0
1	-0.5	0	0	1	0	0
1	-0.3	0	0	0	0	0
1	-0.1	0	0	0	0	0
1	0.1	0	0	0	0	0
1	0.3	0	0	0	0	0
1	0.5	0	1	1	1	0
1	0.7	0	2	1	1	0
1	0.9	0	2	2	1	0
1	1.1	0	3	2	1	0
1	1.3	0	3	3	2	0
1	1.5	0	4	3	2	0
2	-0.9	4	6	5	4	1
2	-0.7	0	4	4	2	0
2	-0.5	0	2	2	1	0
2	-0.3	0	1	1	0	0
2	-0.1	0	0	0	0	0
2	0.1	0	0	0	0	0
2	0.3	0	0	0	0	0
2	0.5	0	0	0	0	0
2	0.7	0	0	0	0	0
2	0.9	0	0	1	0	0
2	1.1	0	0	1	0	0
2	1.3	0	0	0	0	0
2	1.5	0	0	0	0	0
3	-0.9	1	2	1	0	0
3	-0.7	0	0	0	0	0
3	-0.5	0	0	1	0	0
3	-0.3	0	0	0	0	0
3	-0.1	0	0	0	0	0
3	0.1	0	0	0	0	0
3	0.3	0	0	0	0	0
3	0.5	0	1	1	1	0
3	0.7	0	1	1	2	1
3	0.9	0	1	2	2	1
3	1.1	0	1	2	3	2
3	1.3	0	2	3	3	3
3	1.5	0	2	3	4	3

ELEMENT	DEVIATION	SIGNATURES				
4	-0.9	0	4	5	6	8
4	-0.7	0	2	4	4	4
4	-0.5	0	1	2	2	2
4	-0.3	0	0	1	1	0
4	-0.1	0	0	0	0	0
4	0.1	0	0	0	0	0
4	0.3	0	0	0	0	0
4	0.5	0	0	0	0	0
4	0.7	0	0	0	0	0
4	0.9	0	0	1	0	0
4	1.1	0	0	1	0	0
4	1.3	0	0	0	0	0
4	1.5	0	0	0	0	0
5	-0.9	0	0	0	0	0
5	-0.7	0	0	0	0	0
5	-0.5	0	0	0	0	0
5	-0.3	0	0	0	0	0
5	-0.1	0	0	0	0	0
5	0.1	0	0	0	0	0
5	0.3	0	0	0	0	0
5	0.5	0	0	0	0	0
5	0.7	0	0	0	0	0
5	0.9	0	0	0	0	0
5	1.1	0	0	0	0	0
5	1.3	0	0	0	0	0
5	1.5	0	0	0	0	0

TIME= 10.39 SEC

```
      INTEGER A,DASH
      DIMENSION KAM(12,20),IS(20),Y(20),W(20),A(12,20),WV(8,20),ISD(20),
     1   ISN(20),LINE(100),DEN(20),PNU(20),TITLE(2,8),NAME(6,10),YA(20)
     2   ,PCV(150),IE(150)
      COMMON ITF(2),ISIG(2,150,25),NTREE(900,20)
  250 FORMAT(/ 5X,6H TIME=,F6.2,4H SEC)
      READ (5,49) BLANK,PLUS,DASH
   49 FORMAT (3A1)
      DO 45 J=1,100
   45 LINE(J)=DASH
      READ (5,54) ((TITLE(I,J),J=1,8),I=1,2)
   54 FORMAT (16A1)
      READ (5,55) ((NAME(I,J),J=1,10),I=1,6)
   55 FORMAT (60A1)
 1100 CLOCK=ITIME(0)*.01
      TIME=ITIME(0)*.01-CLOCK
      IA=0
      C=-1.1
      JA=1
      READ (5,50) NI,MB,IGR,IPORT
   50 FORMAT(4I10)
      IF (NI.EQ.0) GO TO 1111
      PRINT  59
   59 FORMAT (1H1,30X,'INPUT DATA'////)
      PRINT 7,NI,MB,IGR
      ITERM=2*NI-1
    7 FORMAT (/10X,'NODES=',I2,5X,'EDGES=',I2,5X,'GROUND=',I2///,10X,
     1'EDGE NUMBER',5X,'NODE NUMBERS',5X,'ADMITTANCE',5X,'POWER'/)
      DO 60 I=1,NI
      DO 60 J=1,MB
   60 A(I,J)=0
      JS=1
      DO 61 J=1,MB
      READ (5,51) LE,NX,NY,Y(LE),IS(LE)
   51 FORMAT (I10,2I3,F10.4,I5)
      PRINT 6,LE,NX,NY,Y(LE),IS(LE)
    6 FORMAT (6X,I10,10X,2I6,2X,F12.4,6X,I4)
      A(NX,LE)=1
      A(NY,LE)=1
      YA(J)=Y(J)
      IF (IS(LE)) 62,61,61
   62 JS=JS+1
   61 CONTINUE
  100 FORMAT (//10X,'A MATRIX'/)
      WRITE (6,100)
      DO 101 I=1,NI
  101 WRITE(6,102) (A(I,J),J=1,MB)
  102 FORMAT (13X,25I3)
```

```
      N1=1
      N2=2
      IF (IGR.EQ.1) GO TO 205
      N3=3
      N4=4
      GO TO 210
  205 N3=3
      N4=3
  210 MF=6
  230 DO 20 L=1,MF
      GO TO (12,13,14,15,16,17),L
   12 I1=N1
      I2=N3
      CALL MODNET (I1,I2,A,NI,MB,KAM,N)
      GO TO 200
   13 I1=N2
      I2=N4
      CALL MODNET (I1,I2,A,NI,MB,KAM,N)
      GO TO 200
   14 I1=N1
      I2=N2
      CALL MODNET(I1,I2,A,NI,MB,KAM,N)
      GO TO 200
   15 I1=N3
      I2=N4
      IF (I1.EQ.I2) GO TO 42
      CALL MODNET (I1,I2,A,NI,MB,KAM,N)
      GO TO 200
   42 N=1
      DO 72 I=1,ITERM
   72 WV(L,I)=0.0
      GO TO 20
   16 I1=N1
      I2=N4
      CALL MODNET(I1,I2,A,NI,MB,KAM,N)
      GO TO 200
   17 I1=N2
      I2=N3
      CALL MODNET(I1,I2,A,NI,MB,KAM,N)
      GO TO 200
  200 CONTINUE
      CALL        TREE (N,MB,KAM,IS,JS,Y,NI,W)
      DO 66 I=1,ITERM
   66 WV(L,I)=W(I)
   20 CONTINUE
      IF (IA) 68,69,68
   69 PRINT 67
   67 FORMAT (1H1,20X,'VOLTAGE TRANSFER FUNCTIONS'///)
```

```
68 DO 86 I=1,ITERM
   PNU(I)=0.5*(WV(5,I)+WV(6,I)-WV(3,I)-WV(4,I))
86 DEN(I)=WV(1,I)
   LK=1
   CALL RESULT(DEN,PNU,NI,LINE,ISN,ISD,LK,TITLE,BLANK,PLUS,IA,JA,C)
   DO 87 I=1,ITERM
   PNU(I)=0.5*(WV(5,I)+WV(6,I)-WV(3,I)-WV(4,I))
87 DEN(I)=WV(2,I)
   LK=2
   CALL RESULT(DEN,PNU,NI,LINE,ISN,ISD,LK,TITLE,BLANK,PLUS,IA,JA,C)
   IA=IA+1
   GO TO 300
302 C=-1.1
300 C=C+0.2
   IF (C.GT.1.5) GO TO 301
   Y(JA)=YA(JA)+C*YA(JA)
   PCV(IA)=C
   IE(IA)=JA
   GO TO 230
301 Y(JA)=YA(JA)
   JA=JA+1
   IF (JA.GT.MB) GO TO 700
   GO TO 302
700 IA=IA-1
   DO 701 J=1,2
   WRITE (6,702) (TITLE(J,L),L=1,5)
702 FORMAT (1H1,10X,'FAULT ISOLATION SIGNATURES FOR ',5A1///)
   WRITE (6,705)
705 FORMAT (10X,'ELEMENT',5X,'DEVIATION',10X,'SIGNATURES'//)
   M=ITF(J)
   DO 703 L=1,IA
703 WRITE (6,704) IE(L),PCV(L),(ISIG(J,L,K),K=1,M)
704 FORMAT (13X,I2,10X,F5.1,10X,25I3)
701 CONTINUE
1000 TIME=ITIME(0)*.01-CLOCK
   WRITE(6,250) TIME
   GO TO 1100
1111 STOP
   END
```

```
      SUBROUTINE     MODNET (I1,I2,A,NI,MB,KAM,N)
      INTEGER A
      DIMENSION A(12,20),KAM(12,20)
      DO 4 I =1,NI
      DO 4 J =1,MB
    4 KAM(I,J)=A(I,J)
      DO 3 J=1,MB
      KAM(I1,J)=A(I1,J)+A(I2,J)
      KAM(I2,J)=A(NI,J)
      IF (KAM(I1,J).LE.1) GO TO 3
      KAM(I1,J)=0
    3 CONTINUE
      N=NI-1
      RETURN
      END


      SUBROUTINE  TREE (N,MB,KAM,IS,JS,Y,NI,W)
      DIMENSION KEY(12),KS(12,20),KAM(12,20),LTMP(12,20),CONTL(12),
     1   IS(20),W(20),Y(20),KA(12,12,20)
      COMMON ITF(2),ISIG(2,150,25),NTREE(900,20)
      ITERM=2*NI-1
      DO 58 K=1,ITERM
   58 W(K)=0.0
      NV=N-1
      KSUM=0
      DO 59 I=1,NV
      DO 59 J=1,MB
   59 KA(1,I,J)=KAM(I+1,J)
      DO 60 J=1,MB
   60 KS(1,J)=KAM(1,J)
      NX1=N-1
      NX2=N-2
      NX3=N-3
      NX4=N-4
      NX5=N-5
      NX6=N-6
      NX7=N-7
      NX8=N-8
      NX9=N-9
      NX10=N-10
      NX11=N-11
      CONTL(1)=1.0
      DO 100 I1=1,NX1
      NX=NX1
      M=1
```

```
IF (CONTL(M).EQ.0.0) GO TO 300
CALL         NPART(M,MB,KEY,KS,KA,LTMP,CONTL,NX)
IF (NX2.EQ.0) GO TO 190
IF (CONTL(M+1).EQ.0.0) GO TO 300
DO  99 I2=1,NX2
NX=NX2
M=2
IF (CONTL(M).EQ.0.0) GO TO 100
CALL         NPART(M,MB,KEY,KS,KA,LTMP,CONTL,NX)
IF (NX3.EQ.0) GO TO 190
IF (CONTL(M+1).EQ.0.0) GO TO 100
DO  98 I3=1,NX3
NX=NX3
M=3
IF (CONTL(M).EQ.0.0) GO TO  99
CALL         NPART(M,MB,KEY,KS,KA,LTMP,CONTL,NX)
IF (NX4.EQ.0) GO TO 190
IF (CONTL(M+1).EQ.0.0) GO TO 99
DO  97 I4=1,NX4
NX=NX4
M=4
IF (CONTL(M).EQ.0.0) GO TO 98
CALL         NPART(M,MB,KEY,KS,KA,LTMP,CONTL,NX)
IF (NX5.EQ.0) GO TO 190
IF (CONTL(M+1).EQ.0.0) GO TO 98
DO 96 I5 =1,NX5
NX=NX5
M=5
IF (CONTL(M).EQ.0.0) GO TO 97
CALL         NPART(M,MB,KEY,KS,KA,LTMP,CONTL,NX)
IF (NX6.EQ.0) GO TO 190
IF (CONTL(M+1).EQ.0.0) GO TO 97
DO 95 I6 =1,NX6
NX=NX6
M=6
IF (CONTL(M).EQ.0.0) GO TO 96
CALL         NPART(M,MB,KEY,KS,KA,LTMP,CONTL,NX)
IF (NX7.EQ.0) GO TO 190
IF (CONTL(M+1).EQ.0.0) GO TO 96
DO 94 I7 =1,NX7
M=7
IF (CONTL(M).EQ.0.0) GO TO 95
NX=NX7
CALL         NPART(M,MB,KEY,KS,KA,LTMP,CONTL,NX)
IF (NX8.EQ.0) GO TO 190
IF (CONTL(M+1).EQ.0.0) GO TO 95
DO 93 I8 =1,NX8
NX=NX8
```

```
      M=8
      IF (CONTL(M).EQ.0.0) GO TO 94
      CALL        NPART(M,MB,KEY,KS,KA,LTMP,CONTL,NX)
      IF (NX9.EQ.0) GO TO 190
      IF (CONTL(M+1).EQ.0.0) GO TO 94
      DO 92 I9 =1,NX9
      NX=NX9
      M=9
      IF (CONTL(M).EQ.0.0) GO TO 93
      CALL        NPART(M,MB,KEY,KS,KA,LTMP,CONTL,NX)
      IF(NX10.EQ.0) GO TO 190
      IF (CONTL(M+1).EQ.0.0) GO TO 93
      DO 91 I10=1,NX10
      NX=NX10
      M=10
      IF (CONTL(M).EQ.0.0) GO TO 92
      CALL        NPART(M,MB,KEY,KS,KA,LTMP,CONTL,NX)
      IF(NX11.EQ.0) GO TO 190
      IF (CONTL(M+1).EQ.0.0) GO TO 92
      DO 90 I11=1,NX11
      NX=NX11
      M=11
      IF (CONTL(M).EQ.0.0) GO TO 91
      CALL        NPART(M,MB,KEY,KS,KA,LTMP,CONTL,NX)
190   CONTINUE
      NTR=1
      DO 105 I=1,NV
105   NTR=NTR*KEY(I)
      CALL        XTREE(KEY,NTR,NV,MB,LTMP)
      DO 206 I=1,NTR
      V=1.0
      JJ=0
      DO 205 J=1,MB
      IF (NTREE(I,J).EQ.0) GO TO 205
      V=V*Y(J)
      JJ=JJ+IS(J)
  205 CONTINUE
      W(JJ+JS)=V+W(JJ+JS)
  206 CONTINUE
      GO TO (100,99,98,97,96,95,94,93,92,91,90,89,88,87),M
   87 CONTINUE
   88 CONTINUE
   89 CONTINUE
  90  CONTINUE
  91  CONTINUE
  92  CONTINUE
  93  CONTINUE
   94 CONTINUE
```

```
   95 CONTINUE
   96 CONTINUE
   97 CONTINUE
   98 CONTINUE
   99 CONTINUE
  100 CONTINUE
  300 CONTINUE
      RETURN
      END


      SUBROUTINE NPART(M,MB,KEY,KS,KA,LTMP,CONTL,NX)
      DIMENSION KEY(12),KS(12,20),KA(12,12,20),LTMP(12,20),
     1 CONTL(12)
      KEY(M)=0
      KEND=0
      KCPR=0
      NTEST=0
      DO 2  J=1,MB
      K=J
      IF  (KS(M,J).EQ.0)  GO TO 2
      GO TO 9
    2 CONTINUE
    9 DO 3 I=1,NX
      IK=I
      IF(KA(M,I,K).EQ.0)GO TO 3
      GO TO 15
    3 CONTINUE
   15 DO 72  J=1,MB
      LTMP(M,J)=KS(M,J)*KA(M,IK,J)
      KEY(M)=KEY(M)+LTMP(M,J)
      KA(M,IK,J)=KA(M,IK,J)-LTMP(M,J)
      KS(M,J)=KS(M,J)-LTMP(M,J)
      KS(M+1,J)=KS(M,J)+KA(M,IK,J)
      KEND=KEND+KS(M,J)
      KCPR=KCPR+KA(M,IK,J)
      NTEST=NTEST+KS(M+1,J)
   72 CONTINUE
      DO 12  J=1,MB
      ITEMP=KA(M,NX,J)
      KA(M,NX,J)=KA(M,IK,J)
   12 KA(M,IK,J)=ITEMP
      K=NX-1
      DO 66  I=1,K
      DO 66  J=1,MB
   66 KA(M+1,I,J)=KA(M,I,J)
```

```
      CONTL(M)=KEND*KCPR
      CONTL(M+1)=NTEST
      RETURN
      END


      SUBROUTINE XTREE(KEY,NTR,NV,MB,LTMP)
      DIMENSION LTMP(12,20),KEY(12)
      COMMON ITF(2),ISIG(2,150,25),NTREE(900,20)
      DO 20 I=1,NTR
      DO 20 J=1,MB
20    NTREE(I,J)=0
      MP=1
      DO 26 I=1,NV
      MP=MP*KEY(I)
      MQ=NTR/MP
      MS=MQ*KEY(I)
      M=1
      DO 26 J=1,MB
      IF(LTMP(I,J)-1)26,30,26
30    DO 28 IM=M,NTR,MS
      DO 28 IJ=1,MQ
      IS=IM+IJ-1
28    NTREE(IS,J)=1
      M=M+MQ
26    CONTINUE
      RETURN
      END


      SUBROUTINE RESULT(DEN,PNU,NI,LINE,ISN,ISD,LK,TITLE,BLANK,PLUS,IA,
     1   JA,C)
      DIMENSION DEN(20),PNU(20),LINE(100),ISN(20),ISD(20),TITLE(2,8),
     1   PSIGN(20),QSIGN(20),PPNU(20),PNUR(20),PNUI(20),DDEN(20),
     2   DENR(20),DENI(20),COF(20),WA(25),WB(2,25),B(20),H(2,25),HA(25),
     3   WC(2,25)
      COMMON ITF(2),ISIG(2,150,25),NTREE(900,20)
      ITERM=2*NI-1
      DO 19 J=1,20
      PPNU(J)=0.0
 19   DDEN(J)=0.0
 28   DO 29 J=1,ITERM
      IF (ABS(PNU(J)).GE.1000.) GO TO 30
      IF (ABS(DEN(J)).GE.1000.) GO TO 30
```

```
29 CONTINUE
   GO TO 32
30 DO 31  J=1,ITERM
   PNU(J)=PNU(J)/2.
   DEN(J)=DEN(J)/2.
31 CONTINUE
   GO TO 28
32 CONTINUE
   K=0
   I=0
   ESP=1.E-4
   DO 33  J =1,ITERM
   IF(ABS(DEN(J)).LT.ESP) GO TO 34
   K=K+1
   DEN(K)=DEN(J)
   ISD(K)=J
   IF  (K.EQ.1)  GO TO 11
   IF  (DEN(K))  11,12,12
11 QSIGN(K)=BLANK
   GO TO 34
12 QSIGN(K)=PLUS
34 IF(ABS(PNU(J)).LT.ESP)  GO TO 33
   I=I+1
   PNU(I)=PNU(J)
   ISN(I)=J
   IF  (I.EQ.1)  GO TO 18
   IF  (PNU(I))  18,17,17
18 PSIGN(I)=BLANK
   GO TO 33
17 PSIGN(I)=PLUS
33 CONTINUE
   IF  (ISD(1)-ISN(1))  20,20,21
20 M=ISD(1)
   GO TO 25
21 M=ISN(1)
25 DO 36  J =1,K
36 ISD(J)=ISD(J)-M
   DO 37  J=1,I
37 ISN(J)=ISN(J)-M
   IF  (IA)  200,38,200
38 PRINT  72,((PSIGN(J),PNU(J),ISN(J)),J=1,I)
   PRINT  73,(TITLE(LK,J),J=1,8),(LINE(M),M=1,100)
   PRINT  74,((QSIGN(J),DEN(J),ISD(J)),J=1,K)
72 FORMAT(//20X,10(A1,F6.2,'S**'I2))
73 FORMAT (10X,120A1)
74 FORMAT(20X,10(A1,F6.2,'S**'I2))
   II=ISN(I)
   KK=ISD(K)
```

```
      DO 301 J=1,I
      L=ISN(J)+1
  301 PPNU(L)=PNU(J)
      DO 302 J=1,K
      L=ISD(J)+1
  302 DDEN(L)=DEN(J)
      CALL PCLRT(PPNU,COF,II,PNUR,PNUI,IERN)
      CALL PCLRT(DDEN,COF,KK,DENR,DENI,IERD)
      DO 501 J=1,II
  501 WA(J)=SQRT(PNUR(J)**2+PNUI(J)**2)
      DO 502 J=1,KK
  502 WA(J+II)=SQRT(DENR(J)**2+DENI(J)**2)
      II=II+KK
      KK=II-1
      DO 601 J=1,II
      DO 602 M=1,KK
      IF (WA(M).LE.WA(M+1)) GO TO 602
      ESP=WA(M+1)
      WA(M+1)=WA(M)
      WA(M)=ESP
  602 CONTINUE
  601 CONTINUE
      IF (WA(1).EQ.0.0) GO TO 803
      DO 801 J=1,KK
      IF (WA(J).NE.WA(J+1)) GO TO 801
      WA(J)=0.0
  801 CONTINUE
      L=0
      DO 802 J=1,II
      IF (WA(J).EQ.0.0) GO TO 802
      L=L+1
      WB(LK,L)=WA(J)
  802 CONTINUE
      GO TO 806
  803 WB(LK,1)=0.0
      DO 804 J=1,KK
      IF (WA(J).NE.WA(J+1)) GO TO 804
      WA(J)=0.0
  804 CONTINUE
      L=1
      DO 805 J=1,II
      IF (WA(J).EQ.0.0) GO TO 805
      L=L+1
      WB(LK,L)=WA(J)
  805 CONTINUE
  806 CONTINUE
      M=0
      II=L-1
```

```
      DO 101 J=1,II
      IF (WB(LK,J).EQ.0.0) GO TO 101
      IF (WB(LK,J+1)/WB(LK,J).LE.2.0) GO TO 101
      IF (WB(LK,J+1)/WB(LK,J).LE.10.0) GO TO 102
      IF (WB(LK,J+1)/WB(LK,J).LE.100.0) GO TO 103
      M=M+1
      WB(LK,L+M)=WB(LK,J)*10.
      M=M+1
      WB(LK,L+M)=WB(LK,J+1)/10.0
      M=M+1
      WB(LK,L+M)=(WB(LK,J)+WB(LK,J+1))/2.0
      GO TO 101
  103 M=M+1
      WB(LK,L+M)=WB(LK,J)*2.0
      M=M+1
      WB(LK,L+M)=WB(LK,J+1)/2.0
      GO TO 101
  102 M=M+1
      WB(LK,L+M)=(WB(LK,J)+WB(LK,J+1))/2.0
  101 CONTINUE
      M=M+1
      WB(LK,L+M)=WB(LK,1)/10.0
      M=M+1
      WB(LK,L+M)=WB(LK,L)*2.0
      L=L+M
      M=L-1
      DO 104 KK=1,L
      DO 105 J=1,M
      IF (WB(LK,J).LE.WB(LK,J+1)) GO TO 105
      ESP=WB(LK,J+1)
      WB(LK,J+1)=WB(LK,J)
      WB(LK,J)=ESP
  105 CONTINUE
  104 CONTINUE
      M=0
      KK=L-1
      DO 106 J=1,KK
      IF (WB(LK,J).EQ.0.0) GO TO 106
      IF (WB(LK,J+1)/WB(LK,J).LT.1.01) GO TO 106
      M=M+1
      WC(LK,M)=WB(LK,J)
  106 CONTINUE
      M=M+1
      WC(LK,M)=WB(LK,L)
      ITF(LK)=M
  200 DO 210 J=1,20
  210 B(J)=0.0
      L=ITF(LK)
```

```
      DO 220 M=1,L
      XR=0.0
      XI=0.0
      YR=0.0
      YI=0.0
      DO 230 J=1,I
      B(J)=PNU(J)*WC(LK,M)**ISN(J)
      IF (ISN(J).EQ.0) GO TO 4
      N=ISN(J)
      GO TO (1,2,3,4,1,2,3,4,1,2,3,4,1,2,3,4,1,2,3),N
    1 XI=XI+B(J)
      GO TO 230
    2 XR=XR-B(J)
      GO TO 230
    3 XI=XI-B(J)
      GO TO 230
    4 XR=XR+B(J)
  230 CONTINUE
      DO 240 J=1,K
      B(J)=DEN(J)*WC(LK,M)**ISD(J)
      IF (ISD(J).EQ.0) GO TO 8
      N=ISD(J)
      GO TO (5,6,7,8,5,6,7,8,5,6,7,8,5,6,7,8,5,6,7),N
    5 YI=YI+B(J)
      GO TO 240
    6 YR=YR-B(J)
      GO TO 240
    7 YI=YI-B(J)
      GO TO 240
    8 YR=YR+B(J)
  240 CONTINUE
      IF (IA) 221,222,221
  221 HA(M)=20.0*ALOG10(SQRT(XR**2+XI**2)/SQRT(YR**2+YI**2))
      GO TO 220
  222 H(LK,M)=20.0*ALOG10(SQRT(XR**2+XI**2)/SQRT(YR**2+YI**2))
  220 CONTINUE
      IF (IA.EQ.0) GO TO 270
      DO 40 J=1,M
      HA(J)=ABS(H(LK,J)-HA(J))
      IF (HA(J).LT.0.5) GO TO 41
      IF (HA(J).LT.1.0) GO TO 42
      IF (HA(J).LT.1.5) GO TO 43
      IF (HA(J).LT.2.0) GO TO 44
      IF (HA(J).LT.3.0) GO TO 45
      IF (HA(J).LT.4.0) GO TO 46
      IF (HA(J).LT.5.0) GO TO 47
      IF (HA(J).LT.6.0) GO TO 48
      IF (HA(J).LT.8.0) GO TO 49
```

```
      ISIG(LK,IA,J)=9
      GO TO 40
   49 ISIG(LK,IA,J)=8
      GO TO 40
   48 ISIG(LK,IA,J)=7
      GO TO 40
   47 ISIG(LK,IA,J)=6
      GO TO 40
   46 ISIG(LK,IA,J)=5
      GO TO 40
   45 ISIG(LK,IA,J)=4
      GO TO 40
   44 ISIG(LK,IA,J)=3
      GO TO 40
   43 ISIG(LK,IA,J)=2
      GO TO 40
   42 ISIG(LK,IA,J)=1
      GO TO 40
   41 ISIG(LK,IA,J)=0
   40 CONTINUE
      GO TO 299
  270 DO 271 J=1,M
  271 WC(LK,J)=WC(LK,J)/6.28318
      WRITE(6,250)
  250 FORMAT(/10X,'TEST FREQUENCY (HZ)',10X,'GAIN  (DB)'//)
      PRINT 260,((WC(LK,J),H(LK,J)),J=1,M)
  260 FORMAT (14X,E10.3,15X,E10.3/)
      DO 272 J=1,M
  272 WC(LK,J)=WC(LK,J)*6.28318
  299 CONTINUE
      RETURN
      END
```

BIBLIOGRAPHY

1. Seshu, S. and Waxman, R., "Fault Isolation in Convential Linear Systems---A Feasibility Study", IEEE Transactions, v. R-15, p. 11-16, May 1966.

2. Tsiang, S. H., Haugk, G., and Seckler, H. N., "Maintenance of a Large Electronic Switching System", IEEE Transactions, v. COM-17, p. 1-9, February 1969.

3. Chang, W. T., Determination of Network Functions by Topological Formula, Master's Thesis, Naval Postgraduate School, Monterey, California, June 1968.

4. Van Valkenburg, M. E., Network Analysis, 2d ed., Prentice-Hall, 1964.

5. Kuo, F. F., Network Analysis and Synthesis, 2d ed., Wiley, 1962.

6. Van Valkenburg, M. E., Introduction to Modern Network Sythesis, Wiley, 1960.

7. D'Azzo, J. J. and Houpis, C. H., Feedback Control System Analysis and Synthesis, 2d ed., McGraw-Hill, 1966.

8. Courville, J. D., A Topological Approach to Fault Isolation, Master's Thesis, Naval Postgraduate School, Monterey, California, June 1969.

9. Cave, W. C., "Move Toward Automated Circuit Design", Electronic Design, v. 17, p. 184-189, 15 August 1968.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	20
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Naval Electronics Systems Command Department of the Navy Washington, D. C. 20360	1
4.	Professor S. G. Chan Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	10
5.	Lieutenant (junior grade) James R. Dyer, USN 525 South Hill Street Fayetteville, Arkansas 72701	1

Unclassified
Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

1. ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

3. REPORT TITLE

On Fault Identification of Linear Networks

4. DESCRIPTIVE NOTES (Type of report and inclusive dates)

Master's Thesis; June 1969

5. AUTHOR(S) (First name, middle initial, last name)

James R. Dyer

6. REPORT DATE	7a. TOTAL NO. OF PAGES	7b. NO. OF REFS
June 1969	59	9

8a. CONTRACT OR GRANT NO.	9a. ORIGINATOR'S REPORT NUMBER(S)
b. PROJECT NO.	
c.	9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)
d.	

10. DISTRIBUTION STATEMENT

Distribution of this document is unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY
	Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

A study has been made on the possibility of identifying faulty components of a network by making tests only on external points of that network. The conclusion is made that this is possible, and that a practical method of doing this can be developed from the computer program presented as a result of this investigation.

The method used is to select the voltage transfer functions of a network as the quantities on which the tests will be made. The poles and zeros of these functions are used to select a set of test frequencies. From the measurements made at these frequencies, a set of signatures is available which allow the faulty components to be identified.

DD FORM 1 NOV 65 1473 (PAGE 1)

S/N 0101-807-6811

Unclassified
Security Classification

A-31408

14 KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Fault Identification						
Linear network						